From the Process Hitting to Petri Nets and Back

Loïc Paulevé, Morgan Magnin, Olivier Roux

From the Process Hitting to Petri Nets and Back

Loïc Paulevé[1], Morgan Magnin[2], Olivier Roux[2]
[1] ETH Zürich, Switzerland.
loic.pauleve@ens-cachan.org
[2] LUNAM Université, École Centrale de Nantes, IRCCyN UMR CNRS 6597
(Institut de Recherche en Communications et Cybernétique de Nantes)
1 rue de la Noë - B.P. 92101 - 44321 Nantes Cedex 3, France.

Technical Report - 17th October 2012

Abstract

We define encodings of a Process Hitting into a Petri Net and conversely. It results notably that

- the Process Hitting is a sub-class of Petri Nets;
- any safe Petri Net can be encoded into a weakly-bisimilar Process Hittings with 3 priority classes.

We finally discuss the use of the encoding of a Petri Net into a Process Hitting for providing efficient static analysis of dynamical properties.

1 Main Definitions

Process Hitting (Paulevé, Magnin and Roux, 2012).

Definition 1 (Process Hitting). *A* Process Hitting *is a triple* $(\Sigma, L, \mathcal{H})$*:*

- $\Sigma = \{a, b, \dots\}$ *is the finite countable set of sorts,*
- $L = \prod_{a\in\Sigma} L_a$ *is the set of states, with* $L_a = \{a_0 \dots a_{l_a}\}$ *the finite and countable set of processes of sort* $a \in \Sigma$ *and* l_a *a positive integer,* $a \neq b \Rightarrow a_i \neq b_j\ \forall(a_i, b_j) \in L_a \times L_b$*,*
- $\mathcal{H} = \{a_i \rightarrow b_j \Rsh b_k, \cdots \mid (a, b) \in \Sigma^2,\ (a_i, b_j, b_k) \in L_a \times L_b \times L_b, b_j \neq b_k, a = b \Rightarrow a_i = b_j\}$*, is the finite set of actions of priority* k*.*

Transition between two states $s, s' \in L$ *is defined as follows:*

$$s \rightarrow_{(\Sigma,L,\mathcal{H})} s' \overset{\Delta}{\Leftrightarrow} \exists a_i \rightarrow b_j \Rsh b_k \in \mathcal{H}, s[a] = a_i \wedge s[b] = b_j \wedge s'[b] = b_k \wedge \forall c \in \Sigma, c \neq b, s'[c] = s[c]$$

where $s[a]$ *is the process of sort* a *within the state* s*.*

Given an action $h = b_k \rightarrow a_i \Rsh a_j \in \mathcal{H}$, hitter$(h) \overset{\Delta}{=} b_k$, target$(h) \overset{\Delta}{=} a_i$, and bounce$(h) \overset{\Delta}{=} a_j$.

Petri Nets We consider *safe Petri Nets with read arcs* (Vogler, Semenov and Yakovlev, 1998), as formalised in Def. 2.

Definition 2 (Petri nets with read arcs). *A* Petri Net with read arc *is a tuple* (P, T, W, R)*, where:*

- $P = \{p^1, \dots, p^n\}$*, the finite set of places;*
- $T = \{t^1, \dots, t^m\}$*, the finite set of transitions;*
- $W \subseteq P \times T \cup T \times P$*, the set of (ordinary) arcs;*

- *$R \subseteq P \times T$, the set of read arcs.*
- *$(p, t) \in W \Rightarrow (t, p) \notin W$.*

A (safe) marking *$M \in \wp(P)$ is a set of places. Transition between two markings M and M' is defined as follows:*

$$M \rightarrow_{(P,T,W,R)} M' \stackrel{\Delta}{\Leftrightarrow} \exists t \in T, \mathrm{pre}(t) \subset M \wedge M' = (M \setminus {}^\bullet t) \cup t^\bullet$$

where $\mathrm{pre}(t) = \{p \in P \mid (p, t) \in W \cup R\}$, $t^\bullet = \{p \in P \mid (p, t) \in W\}$, *et* ${}^\bullet t = \{p \in P \mid (t, p) \in W\}$.

2 From Process Hitting to Petri Nets

The encoding of the Process Hitting $(\Sigma, L, \mathcal{H})$ into a Petri Net with read arcs is given by $\mathbf{PN}(\Sigma, L, \mathcal{H})$ (Def. 3). Basically, each process is represented by a place and each action in $\mathcal{H}$ by a transition. For each action $h = b_k \rightarrow a_i \Rsh a_j \in \mathcal{H}$, we add the ordinary arcs (a_i, h) and (h, a_j) in W; and if $a_i \neq b_k$, we add the read arc (b_k, h) in R. The marking corresponding to a Process Hitting state $s \in L$ is simply the set of all the processes in s.

Definition 3. *Given a Process Hitting $(\Sigma, L, \mathcal{H})$, $\mathbf{PN}(\Sigma, L, \mathcal{H}) \stackrel{\Delta}{=} (P, T, W, R)$ is the Petri Net such that:*

- *$P = \bigcup_{a \in \Sigma} L_a$;*
- *$T = \mathcal{H}$;*
- *$W = \{(a_i, h) \mid h \in \mathcal{H} \wedge \mathrm{target}(h) = a_i\} \cup \{(h, a_j) \mid h \in \mathcal{H} \wedge \mathrm{bounce}(h) = a_j\}$;*
- *$R = \{(b_k, h) \mid h \in \mathcal{H} \wedge \mathrm{hitter}(h) = b_k \wedge \mathrm{hitter}(h) \neq \mathrm{target}(h)\}$.*

Given $s \in L$, $[\![s]\!] \stackrel{\Delta}{=} \{s[a] \mid a \in \Sigma\}$.

Starting from a marking $[\![s]\!]$, Theorem 1 states the bisimulation relation between $(\Sigma, L, \mathcal{H})$ et $\mathbf{PN}(\Sigma, L, \mathcal{H})$.

Theorem 1. *Given a Process Hitting $(\Sigma, L, \mathcal{H})$ $\forall s, s' \in L, s \rightarrow_{(\Sigma,L,\mathcal{H})} s' \Longleftrightarrow [\![s]\!] \rightarrow_{\mathrm{PN}(\Sigma,L,\mathcal{H})} [\![s']\!]$.*

Proof. From Def. 3, a marking $[\![s]\!]$ satisfies the pre-condition of a transition h in the Petri Net $\mathbf{PN}(\Sigma, L, \mathcal{H})$ if and only if $\mathrm{hitter}(h) \in s$ and $\mathrm{target}(h) \in s$; and applying the transition h replaces $\mathrm{target}(h)$ by $\mathrm{bounce}(h)$ in the marking. This notably implies that $[\![s]\!] \rightarrow_{\mathrm{PN}(\Sigma,L,\mathcal{H})} M \Rightarrow \exists s' \in L : [\![s]\!]' = M$. $\square$

Fig. 1 illustrates this encoding.
Graphical representation of a Process Hitting: sorts are represented by labeled boxes, and processes by circles (ticks are the identifiers of the processes within the sort, for instance, a_0 is the process ticked 0 in the box a). An action (for instance $a_2 \rightarrow b_1 \Rsh b_0$) is represented by a pair of directed arcs, having the hit part (a_2 to b_1) in plain line and the bounce part (b_1 to b_0) in dotted line. A state is represented by the grayed processes ($\langle a_0, b_0, c_1\rangle$ in this example).
Graphical representation of a Petri Net: places are represented by circles; transitions by squares; ordinary arcs by directed edges; read arcs by non-directed edges. The set of places having a token gives the marking of the Petri Net ($\{a_0, b_0, c_1\}$ is this example).

3 From Petri Nets to Process Hitting

Process Hitting with Priorities We first generalise Def. 1 with the Process Hitting with k Priorities (Def. 4): actions are split in k sub-sets $\mathcal{H}^1, \dots, \mathcal{H}^k$. Actions in $\mathcal{H}^n$, $n \in [1; k]$ have priority k; where 1 (resp. k) is the highest (resp. lowest) priority. An action with priority n can be applied only if none action with priority $< n$ is applicable.

Definition 4. *A* Process Hitting with k Priorities *is tuple $(\Sigma, L, \mathcal{H}^1, \dots, \mathcal{H}^k)$ where:*

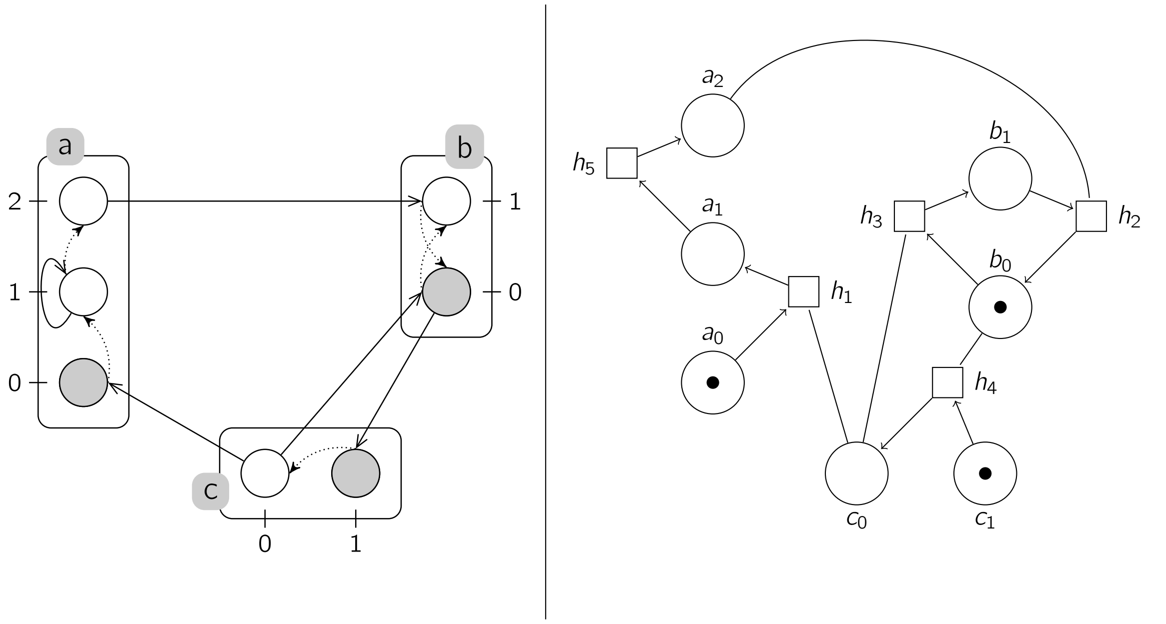

Figure 1: Petri Net (right) of the Process Hitting (left).

- $\Sigma = \{a, b, \dots\}$ *is the finite set of sorts,*
- $L = \prod_{a \in \Sigma} L_a$ *is the set of states, with* $L_a = \{a_0 \dots a_{l_a}\}$ *the finite set of processes of sort* $a \in \Sigma$*, and* l_a *a positive integer.* $a \neq b \Rightarrow a_i \neq b_j \; \forall (a_i, b_j) \in L_a \times L_b$*,*
- $\forall n \in [1; k], \mathcal{H}^n = \{a_i \rightarrow b_j \Rsh b_k, \dots \mid (a, b) \in \Sigma^2 \wedge (a_i, b_j, b_k) \in L_a \times L_b \times L_b \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ *is the set of actions having priority n.*

Given two states $s, s' \in L$*,*

$$s \rightarrow_{(\Sigma, L, \mathcal{H}^1, \dots, \mathcal{H}^k)} s' \iff \exists n \in [1; k] : s \rightarrow_{(\Sigma, L, \mathcal{H}^n)} s'$$
$$\wedge \forall h' \in \mathcal{H}^1 \cup \dots \cup \mathcal{H}^{n-1}, \mathrm{hitter}(h') \notin s \vee \mathrm{target}(h') \notin s \; .$$

Weak-bisimulation of safe Petri Nets The main challenge with the encoding of Petri Nets in Process Hitting is that Process Hitting actions have a very limited scope as their pre-condition is restricted to the test of at most two processes, one of which being the only one to replace.

Hence, a Petri Net transition have to be decomposed into multiple Process Hitting "atomic" actions. In order to warranty the coherence of the sequence of applicable actions in the resulting Process Hitting, we take advantage of their split in priorities; 3 priorities are sufficient here:

- Priority 1: apply the result of an active transition (i.e. de-activate places in the pre-condition and active those in the post-condition);
- Priority 2: compute the transitions that are applicable;
- Priority 3: applicable transition activation.

In particular, our proposed encoding ensures that at most one applicable transition is active at any time.

Given a safe Petri Net with read arcs (P, T, W, R), $\mathbf{PH}(P, T, W, R)$ is the corresponding Process Hitting with 3 Priorities (Def. 5).

Basically, each place $p \in P$ is represented as a dedicated sort having two processes p_0 and p_1, respectively acting for p having none or one token. Each transition $t \in T$ is also represented as a dedicated sort having one process per possible configuration of the places present in the pre-condition; plus one process t_{act} indicating that the transition is active. The actions are then defined as follows:

- Priority 1: for each transition t, the process t_{act} hits the inactive places in its post-condition to make them active, and the active places in its pre-condition (except from read arcs) to make them inactive.
- Priority 2: For each transition t, for each place p in its pre-condition, the process $p_i, i \in \{0, 1\}$ hits every t_σ such that $\sigma[p] \neq p_i$ to make it bounce to the process $t_{\sigma'}$ with $\sigma'[p] = p_i$, and identical to σ for each other place.

  In addition, for each transition t, t_{act} becomes t_σ where σ is the configuration where all places in pre-condition are active.
- Priority 3: For each transition t, if σ is the configuration where all places in pre-condition are active, t_σ hits itself to become t_{act}.

Definition 5 (**PH**(P, T, W, R))**.** **PH**$(P, T, W, R) \stackrel{\Delta}{=} (\Sigma, L, \mathcal{H}^1, \mathcal{H}^2, \mathcal{H}^3, \mathcal{H}^4)$ *is a Process Hitting with* 3 *Priorities corresponding to the safe Petri Net with read arcs* (P, T, W, R)*, where*

- $\Sigma = P \cup T$*;*
- $L = \prod_{p\in P}\{p_0, p_1\} \times \prod_{t\in T}(\{t_\varsigma \mid \varsigma \in \{0, 1\}^{\#\mathrm{pre}(t)}\} \cup \{t_{act}\})$*;*
- $\mathcal{H}^1 = \{t_{act} \rightarrow p_i \Rsh p_j \mid t \in T \wedge ((p \in {}^\bullet t \wedge i = 1 \wedge j = 0) \vee (p \in t^\bullet \wedge i = 0 \wedge j = 1))\}$*;*
- $\mathcal{H}^2 = \{p_i \rightarrow t_\varsigma \Rsh t_{\varsigma'} \mid p \in \mathrm{pre}(t) \wedge i \in \{0, 1\} \wedge \varsigma \in \{0, 1\}^{\#\mathrm{pre}(t)} \wedge \varsigma[p] = p_{1-i} \wedge \varsigma'[p] = p_i \wedge \forall q \in P, q \neq p, \varsigma[q] = \varsigma[q']\} \cup \{t_{act} \rightarrow t_{act} \Rsh t_\varsigma \mid t \in T \wedge \varsigma \in \{1\}^{\#\mathrm{pre}(t)}\}$*;*
- $\mathcal{H}^3 = \{t_\varsigma \rightarrow t_\varsigma \Rsh t_{act} \mid t \in T \wedge \varsigma \in \{1\}^{\#\mathrm{pre}(t)}\}$.

Given a Process Hitting state $s \in L$*,* $[\![s]\!] \in \wp(P)$ *is the corresponding marking:* $[\![s]\!] \stackrel{\Delta}{=} \{s[a] \mid a \in \Sigma\}$.
Given a Petri Net marking $M \in \wp(P)$*,* $(\!|M|\!) = s \in L$ *is the corresponding Process Hitting state, where:* $\forall p \in P, p \in M \Rightarrow s[p] = p_1 \wedge p \notin M \Rightarrow s[p] = p_0$*;* $\forall t \in T, s[t] = t_\varsigma$ *with* $\varsigma \in \{0, 1\}^{\#\mathrm{pre}(t)}$ *and* $\forall p \in \mathrm{pre}(t), p \in M \Rightarrow \varsigma[p] = 1 \wedge p \notin M \Rightarrow \varsigma[p] = 0$.

This construction is linear with the number of places and transitions, but exponential with the cardinality of transition pre-conditions. We notice that factorizing techniques can be used to explode the sort t with 2^n+1 processes, where $n = \#\mathrm{pre}(t)$, into $n-1$ sorts having 4 processes, except one having $4+1$ processes; each of these sorts originating 8 actions. Therefore an equivalent construction can be done in a linear time. We do not detail this trick in this technical report.

Theorem 2. *Given a safe Petri Net with read arcs* (P, T, W, R)*,* $\forall M \in \wp(P)$*,*

1. $\forall M' \in \wp(P)$*,* $M \rightarrow_{(P,T,W,R)} M' \Longleftrightarrow (\!|M|\!) \rightarrow^*_{\mathrm{PH}(P,T,W,R)} (\!|M'|\!)$*;*
2. $\exists s \in L : (\!|M|\!) \rightarrow_{\mathrm{PH}(P,T,W,R)} s \Rightarrow \exists M' \in \wp(P) : s \rightarrow^*_{\mathrm{PH}(P,T,W,R)} (\!|M'|\!)$*;*

where $\rightarrow^*_{\mathrm{PH}(P,T,W,R)}$ *is a finite sequence of* $\rightarrow_{\mathrm{PH}(P,T,W,R)}$.

Proof. Starting from $(\!|M|\!)$, only actions in $\mathcal{H}^3$ are playable, i.e. those activating a transition have its pre-condition satisfied. Playing such an action make one (and only one) t_{act} present, i.e. activates one transition. By construction, all actions in $\mathcal{H}^1$ that are playable can be sequentially and uniquely played, and lead to a unique state where none action in $\mathcal{H}^1$ are playable. These actions have activated and de-activates the post- and pre-conditions (except from read arcs).

Eventually, actions in $\mathcal{H}^2$ are played. By construction, they can be played at most once; and whatever the order of application they all lead to a unique state s, where: for each transition t, $s[t] = t_\varsigma$ where ς is the current configuration of places in the pre-condition of t (including read arcs).

At this point, we remark that $(\!|[\![s]\!]|\!) = s$, hence $(\!|M|\!) \rightarrow^*_{\mathrm{PH}(P,T,W,R)} (\!|M'|\!) \Leftrightarrow M \rightarrow_{(P,T,W,R)} M'$, with $M' = [\![s]\!]$. ☐

Fig. 2 illustrates the encoding of one Petri Net transition into Process Hitting following Def. 5.

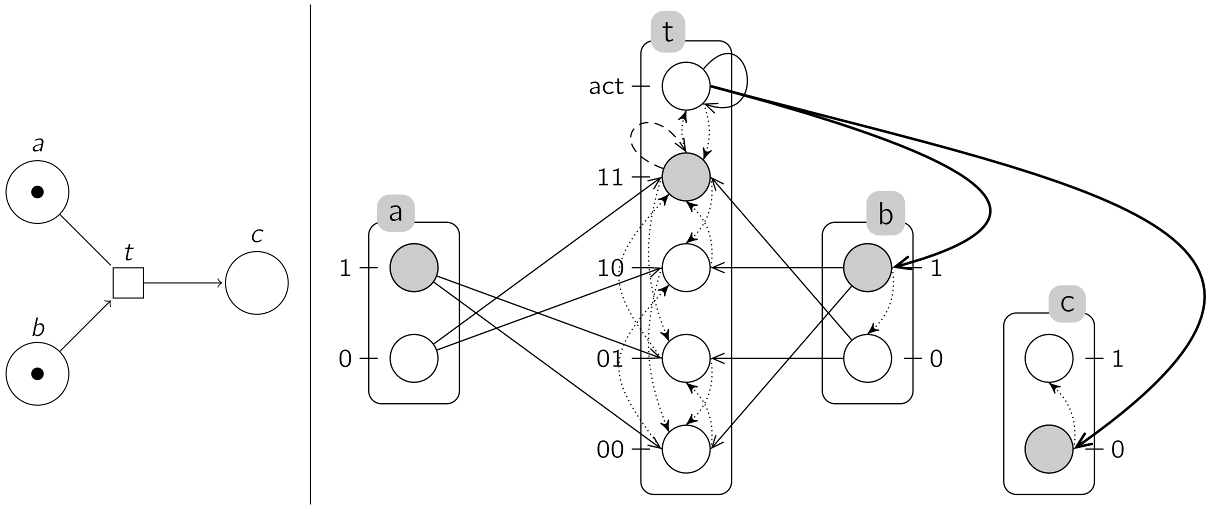

Figure 2: (right) Process Hitting encoding the Petri Net transition (left). Thick actions are in $\mathcal{H}^1$, plain in $\mathcal{H}^2$, and dashed in $\mathcal{H}^3$.

4 Discussion

The Process Hitting as a subclass of Petri Nets The direct construction of a Petri Net from a Process Hitting (Def. 3) indicates that the Process Hitting can be seen as a subclass of Petri Nets. This class of Petri Nets can be informally characterized by the following conditions:

1. all transitions have different pre- and post-conditions, each of them being singletons ($\forall t \in T, \#^\bullet t = \#t^\bullet = 1 \wedge {}^\bullet t \neq t^\bullet$);
2. there exists at most one read arc per transition ($\forall t \in T, \#\{p \in P \mid (p,t) \in R\} \leq 1$);
3. there exists a partition $\Sigma = \{a, b, \dots\}$ of P such that $\forall a \in \Sigma, a \subset P \wedge a \neq \emptyset$; $\cup_{a\in\Sigma} a = P$; $\forall t \in T, \exists a \in \Sigma : {}^\bullet t \cup t^\bullet \subseteq a$; and, $\forall p \in P, \exists a, b \in \Sigma, p \in a \wedge p \in b \Rightarrow a = b$.
4. read arcs conditions are in a different partition than arcs pre- and post-conditions ($\forall t \in T$, let us denote $\Sigma(t)$ the element of Σ such that ${}^\bullet t \cup t^\bullet \subset \Sigma(t)$; $(q,t) \in R \rightarrow q \notin \Sigma(t)$).

The obtained partition is analogous to the PH splitting of processes (here places) within sorts. For each transition $t \in T$ of such a Petri Net, and writing ${}^\bullet t = \{p^i\}$, $t^\bullet = \{p^j\}$, and $\mathrm{pre}(t) = \{p^k, p^i\}$ (with possibly $p^k = p^i$), we have the Process Hitting action $b_k \rightarrow a_i \Rsh a_j$ where $p^k \in b$ and $p^i, p^j \in a$.

By construction, it is clear that applying the **PN** transformation (Def. 3) to the resulting Process Hitting leads to an identical Petri Net. Hence, following Theorem 1, the resulting Process Hitting is bisimilar to the Petri Net with an initial marking containing exactly one place of each partition Σ.

Abstract interpretation of Petri Nets dynamics In (Paulevé et al., 2012), very efficient over- and under-approximations of reachability properties within Process Hittings have been established. They rely on an abstract interpretation designed for Process Hitting dynamics, and make tractable the formal analysis of very large systems (more precisely, Process Hittings having a very large number of sorts, but a few processes per sort).

At the current time, such under-approximations of reachability properties are devoted to Process Hittings without priorities. However, it is worth noticing that for any Process Hitting with k Priorities $(\Sigma, L, \mathcal{H}^1, \dots, \mathcal{H}^k)$, the Process Hitting without priorities $(\Sigma, L, \mathcal{H}^1 \cup \dots \cup \mathcal{H}^k)$ contains all its dynamics.

Hence, Def. 5 gives a straightforward way to efficiently over-approximate reachability properties of safe Petri Nets through the analyses previously mentioned.

References

Paulevé, L., Magnin, M. and Roux, O. (2012). Static analysis of biological regulatory networks dynamics using abstract interpretation, *Mathematical Structures in Computer Science* **22**(04): 651–685.

Vogler, W., Semenov, A. and Yakovlev, A. (1998). Unfolding and finite prefix for nets with read arcs, *in* D. Sangiorgi and R. de Simone (eds), *CONCUR'98 Concurrency Theory*, Vol. 1466 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 501–516.